DIMACS Series in Discrete Mathematics
and Theoretical Computer Science

Relative Size of Certain Polynomial Time Solvable Subclasses of Satisfiability

J. Franco

ABSTRACT. We determine, according to a certain measure, the relative sizes of several well-known polynomially solvable subclasses of SAT. The measure we adopt is the probability that randomly selected k-SAT formulas belong to the subclass of formulas in question. This probability is a function of the ratio r of clauses to variables and we determine those ranges of this ratio that result in membership with high probability.

We show, for any fixed $r > 4/(k(k-1))$, the probability that a random formula is SLUR, q-Horn, extended Horn, CC-balanced, or renamable Horn tends to 0 as $n \to \infty$. We also show that most random unsatisfiable formulas are not members of one of these subclasses.

1. Introduction

The Satisfiability problem (SAT) is to determine whether there exists a satisfying truth assignment for a given Boolean expression. This problem is NP-complete, thus there is no known polynomial-time algorithm for solving it. Because of the importance of SAT in logic, artificial intelligence, and operations research, considerable effort has been spent to determine how to cope with this disappointing reality. Two approaches are: 1) determine whether there exist algorithms for SAT which *usually* present a result in polynomial time; 2) identify special subclasses of SAT that can be solved in polynomial time. This paper is concerned with the second approach.

In this paper we determine, according to a certain measure, the relative sizes of several well-known polynomially solvable subclasses of SAT. The measure we adopt is the probability that randomly selected formulas, drawn from a family of probability spaces, belongs to the subclass of formulas in question. More specifically, the measure is the parameter value on the probability space for which the probability of membership tends to 0 in the limit.

Some notable polynomial time solvable subclasses of SAT (see Section 2 for definitions) are:

1. Horn [**13, 20, 24**],
2. extended Horn [**6**],

1991 *Mathematics Subject Classification.* Primary 68Q15 03B05 68T15.
Supported in part by the Office of Naval Research, N00014-94-1-0382.

3. CC-balanced [**11**],
4. SLUR (Single Lookahead Unit Resolution) solvable [**23**],
5. q-Horn [**3, 4**].

Below, we refer to these as the well-known polynomial time solvable subclasses.

We will not be concerned with various hierarchical subclasses of SAT [**12, 15, 17, 18, 21**], even though portions of them are also solvable in polynomial time. Except for the pure implicational hierarchy [**15**], the best known complexities of these classes is $O(n^k)$ where k reflects the level of a hierarchy; therefore, it is likely that such hierarchies are not efficiently solved for any but the first few levels. Low expressibility is the main factor in ignoring the pure implicational hierarchy. Also 2-CNF is polynomial time solvable [**1, 14**], but we shall not be concerned with random 2-CNF formulas in this paper.

We are interested in the relative sizes of the subclasses above primarily because of the results of Boros *et al.* [**4**], which suggest that the class of q-Horn formulas is close to what might be regarded as the largest easily expressible subclass of SAT that can be solved by a polynomial time, uniform algorithm. They formulate a set of linear constraints, based on the input formula (with n variables) and a real parameter Z, and show that, for any fixed $c > 0$, the class of formulas that satisfies the constraints with $Z = 1 + c\frac{\log n}{n}$ can be solved in polynomial time. In addition, the class that satisfies the constraints with $Z = 1$ is precisely the q-Horn class (see Definition 2.7). On the other hand, for any $\beta < 1$, the class of formulas that satisfy these constraints with $Z = 1 + \frac{1}{n^\beta}$ is NP-complete.

To measure size we use a well-known probability distribution $\mathcal{M}_{\Updownarrow,\backslash,||}$, also known as the constant clause-width model, defined over the sample space of k-CNF formulas, which is defined over a set of n propositional variables. The clause space for $\mathcal{M}_{\Updownarrow,\backslash,||}$ consists of the $2^k\binom{n}{k}$ clauses with k literals such that no two literals are based on the same variable. The formula space consists of all multisets of m clauses. Each multiset has equal probability, which is $\left(2^k\binom{n}{k}\right)^{-m}$. Thus clauses are generated by sampling *without* replacement, while formulas are generated by sampling *with* replacement. This paper restricts attention to $k \geq 3$. Probability spaces will frequently be grouped according to ratio $r \equiv m/n$.

We determine those regions of the parameter space $\langle m, n, k\rangle$ over which a random formula has low probability of being in a certain subclass, such as q-Horn, etc. We use this approach because

1. several of the subclasses considered are incomparable;
2. the ratio $r = m/n$ provides a scale which has been shown, both theoretically and experimentally, to measure the *hardness* of formulas;
3. many results already proven for the formula distribution may be used to add dimension to the results presented here.

Except for point 8, the following results for random formulas under $\mathcal{M}_{\Updownarrow,\backslash,||}$, $k \geq 3$, are known [**5, 7, 8, 9, 10, 16, 19**].

1. For any fixed $r \geq .65\ 2^k$, the probability that a random formula is unsatisfiable tends to 1 as $n \to \infty$.
2. For any fixed $r \geq .65\ 2^k$, there is no known algorithm that will verify unsatisfiability of a random formula in polynomial time with probability tending to 1 as $n \to \infty$.
3. For any fixed $r < .25\ 2^k/k$, the probability that a random formula is satisfiable tends to 1 as $n \to \infty$.

4. For any fixed $r < .25\ 2^k/k$, with probability tending to 1 as $n \to \infty$, a random formula that is satisfiable can be solved in polynomial time by an iterative variable elimination algorithm that relies primarily on choosing variables for elimination from a shortest clause.
5. For any fixed $r < 1.63$, a random 3-CNF formula can be satisfied by repeated application of the pure literal rule, with probability tending to 1 as $n \to \infty$.
6. For any fixed $r < 1$, a random formula can be satisfied by applying any algorithm for 2-SAT to the formula with all but 2 literals randomly removed from each clause, with probability tending to 1 as $n \to \infty$.
7. The average number of occurrences of a variable in a random formula is less than 1 if $r < 1/k$.
8. The average number of cycles in a random formula is bounded from above by a small constant if $r < 1/k^2$. A cycle in this context means an undirected cycle in the graph formed by considering each clause as a node and connecting each pair of clauses that share a variable. This result is proved in Section 3.3.

The first two points above show where random formulas are "hard" and the last six points show where random formulas are "easy." Notice the progression from very hard formulas (not easily solved by resolution) at $r = .65\ 2^k$, r fixed, to usually solvable in polynomial time at $r = .25\ 2^k/k$ by non-trivial heuristics to easily solvable by a 2-SAT algorithm at $r = 1$ to very easily solvable since variables usually occur one time in a formula at $r = 1/k$ to trivially solvable due to no or few cycles at $r = 1/k^2$. Thus, $\mathcal{M}_{\Updownarrow,\backslash,||}$ may be thought of as a generator of formulas of hardness controlled by the ratio r. We wish to see where the well-known classes fall on this scale.

In this paper we present the following result. Definitions appear in Section 2.

- For any fixed $r > 4/(k(k-1))$, the probability that a random formula is SLUR, q-Horn, extended Horn, CC-balanced, or renamable Horn tends to 0 as $n \to \infty$.

Therefore, the well-known polynomial time solvable subclasses of SAT, by our measure, do not represent most "easy" formulas for a wide range of values of r and are much smaller than other classes of formulas that, as an aggregate, are easily solved with high probability. It is interesting to note that the probability that a random formula $\mathcal{F}$ is in one of the well-known subclasses tends to 0 as $n \to \infty$, unless the average number of occurrences of a variable in $\mathcal{F}$ is less than $4/(k-1)$, a very small number. In addition, our results show that most random unsatisfiable formulas are not members of one of the well-known subclasses.

2. Polynomial Time Solvable Subclasses of SAT

We specify SAT for the purposes of this paper as follows. Let $V = \{v_1, ..., v_n\}$ be a set of n Boolean variables. Let $L_n = \{v_1, \bar{v_1}, ..., v_n, \bar{v_n}\}$ be a set of n positive and n negative literals over variables in V. A truth assignment to the literals L is a mapping $t : L \to \{\mathcal{T}, \mathcal{F}\}$ such that $t(\bar{v}) = \mathcal{T}$ if and only if $t(v) = \mathcal{F}$. A subset of literals L is called a clause. A clause C has truth value $\mathcal{T}$ under a truth assignment t if and only some literal in C is assigned $\mathcal{T}$. A collection (multiset) of clauses, $\{C_1, C_2, \ldots, C_m\}$, is a *formula in Conjunctive Normal Form* (CNF). From now on, it is understood that *formula* means formula in Conjunctive Normal Form. A formula $\mathcal{F}$ is satisfiable if and only if there exists a truth assignment t such

that every clause in $\mathcal{F}$ has truth value $\mathcal{T}$ under t. Such a t is said to satisfy $\mathcal{F}$. The objective of an algorithm for SAT is to determine whether a given formula is satisfiable.

It will be useful to represent a formula $\mathcal{F}$ as an $m \times n$ $(0, \pm 1)$-matrix.

DEFINITION 2.1. Given a formula $\mathcal{F}$, its *clause-variable matrix*, denoted as $M_{\mathcal{F}}$, is the $m \times n$ matrix in which element (i, j) has the value $+1$ if clause C_i has literal v_j, has the value -1 if clause C_i has literal $\bar{v}_j$, and has the value 0 otherwise.

The remainder of this section defines certain subclasses of SAT that are solved in polynomial time.

DEFINITION 2.2. A formula $\mathcal{F}$ is *Horn* if and only if every row of $M_{\mathcal{F}}$ has at most one $+1$ value.

Horn formulas can be solved in linear time by unit resolution [**13, 20, 24**].

DEFINITION 2.3. **(Lewis [22])** A formula $\mathcal{F}$ is *renamable Horn* if and only if multiplying each of some subset of columns of $M_{\mathcal{F}}$ by -1 yields an M matrix corresponding to a Horn formula.

Renamable Horn formulas can also be solved in linear time [**2**].

Extended Horn formulas can be expressed as linear inequalities for which 0-1 solutions can always be found (if one exists) by rounding a real solution obtained using an LP relaxation [**6**]. We find an alternative characterization is easier to understand.

DEFINITION 2.4. Given a formula $\mathcal{F}$, let R be a rooted directed tree in which each edge is labeled with a different variable from the set V.

A clause C is an *extended Horn clause w.r.t. R* if the positive literals of C correspond to a (possibly empty) directed path P in R, and the set of negative literals in C correspond to a set of directed paths $N_1, N_2, ..., N_t$ of R, and exactly one of the following conditions holds:

1. $N_1, N_2, ..., N_t$ start at the root s.
2. $N_1, N_2, ..., N_{t-1}$, (say), start at the root s, and N_t starts at a vertex $q \neq s$. Moreover, if P is not empty, it also starts at q.

A formula is an *extended Horn formula w.r.t. R* if each of its clauses is an extended Horn clause w.r.t. R. A formula is an *extended Horn formula* if it is an extended Horn formula w.r.t. some such rooted directed tree R.

One tree R for a given Horn formula is a star (one root and all leaves with an edge for each variable in the formula). Hence, the class of extended Horn formulas is a generalization of the class of Horn formulas.

Unsatisfiable extended Horn formulas can be recognized in polynomial time, by an algorithm based on unit-resolution plus rounding [**6**]. Therefore, if a formula is known to be extended Horn *a priori*, it can be solved in polynomial time. However, there is no known polynomial time algorithm for recognizing extended Horn formulas.

The following class is also rooted in Linear Programming.

DEFINITION 2.5. A formula $\mathcal{F}$ is *CC-balanced* if in every submatrix of $M_{\mathcal{F}}$ with exactly two nonzero entries per row and per column, the sum of the entries is a multiple of four [**25**].

The motivation for studying CC-balanced formulas is the question, for SAT, when do Linear Programming relaxations have integer solutions? CC-balanced formulas can be recognized and solved in polynomial time **[11]**.

The class SLUR, for Single Lookahead Unit Resolution, is peculiar in that it is defined by an algorithm, and not by structural properties of formulas. The algorithm defining SLUR is given below. In it, the function *unitprop*($\mathcal{F}$) returns the result of performing the well-known *unit clause simplification* until no unit clauses remain in the formula. It also returns the set of unit clauses found and derived. It is known that *unitprop* can be implemented in time linear in $|\mathcal{F}|$ **[12]**.

Algorithm **SLUR**($\mathcal{F}$)
Input: A CNF formula $\mathcal{F}$ with no empty clause
Output: A satisfying partial truth assignment for the variables in $\mathcal{F}$, or "unsatisfiable", or "give up"

Initialize $\mathcal{F} :=$ *unitprop*($\mathcal{F}$).
Initialize $t :=$ the set of unit clauses returned by *unitprop*.
If $\emptyset \in \mathcal{F}$, then
    Output "unsatisfiable" and halt.
While $\mathcal{F}$ is not empty do the following:
    Select a variable v appearing as a literal of $\mathcal{F}$.
    Set $\mathcal{F}_\infty :=$ *unitprop*$(\mathcal{F} \cup \{\acute{\sqsubseteq}\})$.
    Set $t_1 :=$ unit clauses returned by *unitprop*.
    Set $\mathcal{F}_\in :=$ *unitprop*$(\mathcal{F} \cup \{\sqsubseteq\})$.
    Set $t_2 :=$ unit clauses returned by *unitprop*.
    If $\emptyset \in \mathcal{F}_\infty$ and $\emptyset \in \mathcal{F}_\in$, then
        Output "give up" and halt.
    Otherwise, if $\emptyset \notin \mathcal{F}_\infty$, then
        Set $\mathcal{F} := \mathcal{F}_\infty$.
        Set $t := t \cup t_1$.
    Otherwise,
        Set $\mathcal{F} := \mathcal{F}_\in$.
        Set $t := t \cup t_2$.
    (Continue the loop.)
Output t.
End Algorithm **SLUR**

DEFINITION 2.6. A formula is in the class SLUR if, for all possible sequences of selected variables, algorithm SLUR does not give up.

Algorithm SLUR takes linear time with the modification, due to Truemper **[25]**, that unit resolution (in *unitprop*) be applied simultaneously to both branches of a selected variable, abandoning one branch if the other finishes first without falsifying a clause. Note that due to the definition of this class, the question of class recognition is avoided. The class SLUR was developed as a generalization of other classes including Horn, renamable Horn, extended Horn, and CC-balanced formulas **[23]**.

The class q-Horn was developed in **[3, 4]**.

DEFINITION 2.7. Let $\{v_1, v_2, ..., v_n\}$ be a set of Boolean variables. For clause C_i, let P_i be the set of indices of its positive literals and let N_i be the set of indices

of its negative literals. Construct the following system of inequalities:

$$\sum_{j\in P_i}\alpha_j + \sum_{j\in N_i}(1-\alpha_j) \quad\leq\quad Z,\ (i=1,2,...,m),\text{ and} \tag{1}$$

$$0\leq\alpha_j\leq 1 \qquad (j=1,2,...,n). \tag{2}$$

where $Z \in R^{+}$. If all these constraints can be satisfied with $Z = 1$, then the formula is q-Horn.

We may also characterize this class as a special case of monotone decomposition of matrices [**25**]. Given formula $\mathcal{F}$, the monotone decomposition of $M_{\mathcal{F}}$, consists of multiplying some columns by -1 and moving rows and columns to form the following partition into submatrices:

$$\left(\frac{A^1 \mid E}{D \mid A^2}\right)$$

where the submatrix A^1 has at most one $+1$ entry per row, the submatrix D contains only -1 or 0 entries, the submatrix E has only 0 entries, and the submatrix A^2 has no restrictions. Below, we will be concerned with the maximum monotone decomposition where matrix A^1 is the largest possible. Maximum monotone decompositions are essentially unique [**25**].

DEFINITION 2.8. If the maximum monotone decomposition of $M_{\mathcal{F}}$ is such that A^2 has no more than two nonzero entries per row, then $\mathcal{F}$ is *q-Horn*.

Recognition of q-Horn formulas is made easy by the fact that monotone decomposition can be carried out in linear ($O(m+n)$) time [**25**]. Once a q-Horn formula $\mathcal{F}$ is in its decomposed form it can be solved in linear time as follows. Treat submatrix A^1 as a Horn formula and solve it in linear time using a method that returns a minimum, unique truth assignment for the formula with respect to *true* [**13, 20**]. If the Horn formula is unsatisfiable, then $\mathcal{F}$ is unsatisfiable. Otherwise, remove all rows satisfied by the unique minimum truth assignment. Solve what is left of submatrix A^2 by a 2-SAT algorithm [**1, 14**]. If a satisfying assignment is found, it may be combined with the unique minimum assignment above to give an assignment satisfying $\mathcal{F}$. Otherwise, $\mathcal{F}$ is not satisfiable.

In the analysis below we will not directly consider some of the classes defined above because they are subclasses of either SLUR or q-Horn. However, the classes of SLUR, and q-Horn formulas are incomparable as the following examples show.

EXAMPLE 2.9. Any formula $\{\bar{v}_1, v_2, v_3\}\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}\ldots$ is not q-Horn. To see this, construct inequalities as in (1) and (2) for the first two clauses. These force $\alpha_1 \geq 2 - Z/2$ which requires $Z \geq 2$.

EXAMPLE 2.10. The formula $\{\bar{v}_1, v_2, v_3\}\{\bar{v}_1, \bar{v}_2, \bar{v}_3\}$ is not q-Horn but it is obviously SLUR. This formula can easily be extended to less trivial SLUR formulas that are not q-Horn.

EXAMPLE 2.11. The formula $\{v_1, v_2, \bar{v}_3\}\{v_1, \bar{v}_2, \bar{v}_4\}\{\bar{v}_1, v_2, \bar{v}_5\}\{\bar{v}_1, \bar{v}_2, \bar{v}_6\}\ldots$, where $\ldots$ is Horn and does not contain v_1 or v_2 is q-Horn but not SLUR.

3. Analysis

We restate the definition of model $\mathcal{M}_{\Updownarrow,\backslash,||}$ in terms of the notation of Section 2. Let $\mathcal{C}_{\backslash,||}$ be the set of all subsets of L_n of size k such that no element of $\mathcal{C}_{\backslash,||}$ contains duplicate or complementary literals. Random formulas generated according to

$\mathcal{M}_{\Updownarrow,\backslash,||}$ contain m clauses selected uniformly, independently, and with replacement from $\mathcal{C}_{\backslash,||}$. We will be interested in the case $k \geq 3$ since random formulas generated from $\mathcal{M}_{\Updownarrow,\backslash,||}$, $k \leq 2$, are solved in linear time by existing 2-SAT algorithms [**1, 14**].

3.1. SLUR Analysis.

DEFINITION 3.1. For any even $x \geq 4$, call a set of x clauses an *equivalence cycle* if all but two literals can be removed from each clause, the variables can be relabeled, and the clauses can be reordered in the following sequence

$$\{v_1, \bar{v}_2\}\{v_2, \bar{v}_3\}\ldots\{v_{\frac{x}{2}}, v_1\}\{\bar{v}_1, \bar{v}_{\frac{x}{2}+1}\}\ldots\{v_{x-1}, \bar{v}_1\},$$

where $v_i \neq v_j$ if $i \neq j$. Given an equivalence cycle $\mathcal{C} \subset \mathcal{F}$, if every clause $C \notin \mathcal{C}$ contains at most $k-2$ literals that are the same as or complementary to the removed literals of $\mathcal{C}$, and no two of the literals removed from $\mathcal{C}$ are the same or complementary, then $\mathcal{C}$ is called a *blocked equivalence cycle.* The variable v_1 is called the *end variable* of the cycle.

LEMMA 3.2. *If a formula $\mathcal{F}$ has a blocked equivalence cycle, then $\mathcal{F}$ is not SLUR.*

Proof: In algorithm SLUR, choose for elimination the variables removed from the blocked equivalence cycle of $\mathcal{F}$; proceed down the search tree in the direction corresponding to falsifying the literals in the blocked equivalence cycle. By hypothesis, there will be no unit clauses or empty clauses and yet what's left will be unsatisfiable due to the equivalence cycle. This violates the definition of SLUR. □

THEOREM 3.3. *Under $\mathcal{M}_{\Updownarrow,\backslash,||}$, the probability that a random formula $\mathcal{F}$ is in the class SLUR tends to 0 if $r > 4c/(k^2-k)$, $c > 1$ a constant.*

Proof: We apply the second moment method to prove the theorem. Let B_i denote the number of blocked equivalence cycles of size i in a random formula $\mathcal{F}$. Let $x = \lceil \ln^2 n \rceil$ or $x = \lceil \ln^2 n \rceil + 1$, whichever is even. We find $E(B_x)$, the expected number of blocked equivalence cycles of size x, and $E(B_x^2)$, the second moment of B_x. We show $E(B_x) = \alpha^x$, for some $\alpha > 1$, when $r > 4/(k^2-k)$. Then we show $E(B_x^2) = E(B_x)^2(1+o(1))$ under the same conditions. Therefore, by Chebyshev's inequality,

$$Pr(B_x = 0) \leq Pr(|B_x - E(B_x)| \geq E(B_x)) \leq \frac{E(B_x^2) - E(B_x)^2}{E(B_x)^2} = o(1)$$

when $r > 4/(k^2-k)$.

First, we find $E(B_x)$. Pick $x \geq 4$ clauses, and $x-1$ variables. Arrange the clauses with variables so as to construct an equivalence cycle where the end variable of the first clause is repeated in the $x/2$th and $x/2+1$th clauses of the cycle. The literal pattern of the two literals of each clause that cause it to be in the equivalence cycle is fixed. The probability that the clauses in the sequence match their patterns is

$$\prod_{i=0}^{x-1}\left(\frac{2^{k-2}\binom{n-x-(k-2)i}{k-2}}{2^k\binom{n}{k}}\right) = \left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^x (n-x)^{\overline{(k-2)x}}$$

where $a^{\overline{b}}$ is used to denote the product $a(a-1)(a-2)\ldots(a-b+1)$. The probability that any non-cycle clause does not have more than $k-2$ literals taken from the

set of $x-1$ chosen variables and their complements is, ignoring insignificant terms, $1-k(x/n)^{k-1}$. Hence, the probability that all cycle clauses match their patterns and non-cycle clauses do not share more than $k-2$ literals with the cycle clauses is

$$\left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^x (n-x)^{\overline{(k-2)x}} \left(1-k\left(\frac{x}{n}\right)^{k-1}\right)^{m-x}.$$

The number of ways to select x clauses is $m(m-1)(m-2)\dots(m-x+1) = m^{\overline{x}}$. The number of ways to choose $x-1$ variables is $n(n-1)(n-2)\dots(n-x+2) = n^{\overline{x-1}}$. Therefore, ignoring insignificant terms for convenience of presentation,

$$E(B_x) \;=\; m^{\overline{x}} n^{\overline{x-1}} \left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^x (n-x)^{\overline{(k-2)x}} \left(1-k\left(\frac{x}{n}\right)^{k-1}\right)^{m-x}.$$

Since $m^{\overline{x}} > m^x(1-x/m)$, $n^{\overline{x-1}} > n^{x-1}(1-(x-1)/n)$, and $(n-x)^{\overline{(k-2)x}} > (n-(k-1)x)^{(k-2)x}$,

$$\begin{aligned} E(B_x) &> \frac{1}{n}\left(\frac{k(k-1)mn(n-(k-1)x)^{k-2}}{4n^k}\right)^x \left(1-\frac{x^2}{n}-\frac{x}{m}-\frac{x}{n}+\dots\right) \\ &> \frac{1}{n}\left(\frac{k(k-1)m}{4n}\left(1-O\left(\frac{k^2x}{n}\right)\right)\right)^x \left(1-\frac{x^2}{n}+\dots\right) \\ &= \frac{1}{n}\left(\frac{k(k-1)m}{4n}\right)^x \left(1-O\left(\frac{k^2x^2}{n}\right)\dots\right). \end{aligned}$$

If $m/n > 4c/(k^2-k)$, where c is any constant greater than 1, $E(B_x) > \frac{1}{n}c^{\ln^2 n} > \alpha^{\ln n}$, in the limit, for some $\alpha > 1$.

Next, we find $E(B_x^2)$. Order all possible patterns of variable choices and clause choices. There are $m^{\overline{x}} n^{\overline{x-1}}$ of these. Let $B_x = X_1 + X_2 + X_3 + \dots$ where each X_i is 1 if, for the ith pattern, there is a blocked equivalence cycle and is 0 otherwise. Then $E(B_x^2) = \sum_{i,j} E(X_i X_j)$.

Suppose patterns i and j have q clauses in common. If $q=0$ it is possible for both patterns to co-exist in $\mathcal{F}$. But, if $q>0$, they may not be able to co-exist: in particular, the variable assignments at the clauses shared by both patterns must agree. The number of different possible variable patterns supporting consistent overlapping cycles is no greater than $n^{2(x-1)-q}$, except for $q=x$ in which case it is n^{x-1}. The probability that patterns i and j have q clauses in commom is

$$\binom{x}{q}\frac{(m-x)^{\overline{x-q}}x^{\overline{q}}}{m^{\overline{x}}}.$$

Given two consistent blocked equivalence cycles, the probability that both are in $\mathcal{F}$ is (ignoring insignificant terms as above) no greater than

$$\begin{aligned} &\prod_{i=0}^{2x-q-1}\left(\frac{2^{k-2}\binom{n-x-(k-2)i}{k-2}}{2^k\binom{n}{k}}\right)\left(1-k\left(\frac{x}{n}\right)^{k-1}\right)^{m-2x+q} \\ = \; &\left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^{2x-q}(n-x)^{\overline{(k-2)(2x-q)}}\left(1-k\left(\frac{x}{n}\right)^{k-1}\right)^{m-2x+q}. \end{aligned}$$

Therefore,

$$\sum_{i,j} E(X_i X_j)$$

$$
\begin{aligned}
&< m^{\overline{2x}} n^{\overline{2(x-1)}} \frac{(m-x)^{\overline{x}}}{m^{\overline{x}}} \left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^{2x} (n-x)^{\overline{(k-2)2x}} \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{m-2x} \\
&\times \Big(\sum_{q=0}^{x-1} \frac{1}{(m-2x)^{\overline{q}}(n-2x)^{\overline{q}}} \binom{x}{q} \frac{(m-x)^{\overline{x-q}} x^{\overline{q}}}{m^{\overline{x}}} \left(\frac{4n^{\overline{k}}}{k(k-1)}\right)^{q} \frac{(n-x)^{\overline{(k-2)(2x-q)}}}{(n-x)^{\overline{(k-2)2x}}} \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{q} \\
&+ \frac{n}{(m-2x)^{\overline{x}}(n-2x)^{\overline{x}}} \frac{x^{\overline{x}}}{m^{\overline{x}}} \left(\frac{4n^{\overline{k}}}{k(k-1)}\right)^{x} \frac{(n-x)^{\overline{(k-2)x}}}{(n-x)^{\overline{(k-2)2x}}} \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{x} \Big) \\
&< m^{\overline{x}} m^{\overline{x}} n^{\overline{x-1}} n^{\overline{x-1}} \left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^{2x} (n-x)^{\overline{(k-2)x}} (n-x)^{\overline{(k-2)x}} \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{m-2x} \\
&\times \left(\sum_{q=0}^{x} \binom{x}{q} \left(\frac{4xn^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}}\right)^{q} + n\left(\frac{4xn^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}}\right)^{x}\right) \\
&< E(B_x)^2 \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{-m} \\
&\times \left(\left(1 + \frac{4xn^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}}\right)^{x} + n\left(\frac{4xn^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}}\right)^{x}\right) \\
&< E(B_x)^2 \left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{-m} \\
&\times \left(1 + \frac{12x^2n^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}} + n\left(\frac{4xn^k}{k(k-1)(m-3x)^2(n-2kx)^{k-1}}\right)^{x}\right) \\
&= E(B_x)^2(1+o(1))(1+o(1/n))
\end{aligned}
$$

since $x^2n^k/(k(k-1)(m-3x)^2(n-2kx)^{k-1}) \to x^2n/(k(k-1)m^2) = O(x^2/n)$ and $n(4xn^k/(k(k-1)(m-3x)^2(n-2kx)^{k-1}))^x \to n(4xn/(k(k-1)m^2))^x = o(1/n^{x-2})$ due to $m-3x \to m$, $n-2kx \to n$, and $m/n = r > 4/(k^2-k)$. □

3.2. Q-Horn Analysis.

DEFINITION 3.4. For $x = \lfloor \ln n \rfloor \geq 4$, call a set of x clauses a *c-cycle* if all but two literals can be removed from each of $x-2$ clauses, all but three literals can be removed from two clauses, the variables can be relabeled, and the clauses can be reordered in the following sequence

$$\{v_1, \bar{v}_2\}\{v_2, \bar{v}_3\}\ldots\{v_i, \bar{v}_{i+1}, v_{x+1}\}\ldots\{v_j, \bar{v}_{j+1}, \bar{v}_{x+1}\}\ldots\{v_x, \bar{v}_1\},$$

where $v_i \neq v_j$ if $i \neq j$. Given a c-cycle $\mathcal{C} \subset \mathcal{F}$, if none of the literals removed from $\mathcal{C}$ are the same or complementary, then $\mathcal{C}$ is called a *q-blocked c-cycle*.

LEMMA 3.5. *If a formula $\mathcal{F}$ has a q-blocked c-cycle then it is not q-Horn.*

Proof: Let a q-blocked c-cycle in $\mathcal{F}$ be represented as follows

$$\{v_1, \bar{v}_2, \ldots\}\ldots\{v_i, \bar{v}_{i+1}, v_{x+1}, \ldots\}\ldots\{v_j, \bar{v}_{j+1}, \bar{v}_{x+1}, \ldots\}\ldots\{v_x, \bar{v}_1, \ldots\}.$$

Develop inequalities (1) and (2) for the formulas above. We get, after rearranging terms in each

$$\alpha_1 \leq Z - 1 + \alpha_2 - \ldots \tag{3}$$

$$\ldots$$

$$\alpha_i \leq Z - 1 + \alpha_{i+1} - \alpha_{x+1} - \ldots$$

$$\ldots$$
$$\alpha_j \le Z - 1 + \alpha_{j+1} - (1 - \alpha_{x+1}) - \ldots$$
$$\ldots$$
$$\alpha_x \le Z - 1 + \alpha_1 - \ldots. \tag{4}$$

From inequalities (3) to (4) we deduce

$$\alpha_1 \le xZ - x + \alpha_1 - (1 - \alpha_x + \alpha_x) - \ldots$$

or

$$0 \le xZ - x - 1 - \ldots$$

where all the terms in $\ldots$ are non-negative. Thus, all solutions to (3) through (4) require $Z > (x+1)/x = 1 + 1/x = 1 + 1/\lfloor \ln^2 n \rfloor > 1 + 1/n^{\beta}$ for any fixed $\beta < 1$. This violates the result of [4] that requires $Z \le 1$ in order for $\mathcal{F}$ to be q-Horn. $\square$

THEOREM 3.6. *Under $\mathcal{M}_{\Updownarrow,\backslash,||}$, the probability that a random formula $\mathcal{F}$ is q-Horn tends to 0 if $r > 4c/(k^2 - k)$, $c > 1$ a constant.*

Proof: This is another application of the second moment method closely following that of Theorem 3.3. This time we seek the expected number of q-blocked c-cycles in $\mathcal{F}$ and to show that this expectation is large and variance small over the indicated range of r. Then, by Lemma 3.5, the result follows.

Taking advantage of the remarkable similarities between q-blocked c-cycles and blocked equivalence cycles, we need modify the proof of Theorem 3.3 only by the small changes due to a slightly different probability of the event being measured and count of the number of possibilities. Thus,

$$m^{\overline{x}} n^{\overline{x-1}} \text{ is replaced by } m^{\overline{x}} n^{\overline{x+1}},$$

and

$$\left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^{x} (n-x)^{\overline{(k-2)x}} \text{ is replaced by}$$

$$\left(\frac{k(k-1)}{4n^{\overline{k}}}\right)^{x-2} \binom{x}{2} \left(\frac{k(k-1)(k-2)}{8n^{\overline{k}}}\right)^{2} (n-x-2)^{\overline{(k-2)(x-2)+2(k-3)}},$$

and the term $\left(1 - k\left(\frac{x}{n}\right)^{k-1}\right)^{2x}$ is not used. The details are omitted. $\square$

3.3. Cycles. As stated before, a cycle in a formula is an undirected cycle in the graph formed by considering each clause as a node and connecting each pair of clauses that share a variable. A formula without a cycle is trivially satisfied by assigning values to variables satisfying the "leaf" clauses, working inward to the root(s). A formula with no cycles is a member of all the well known polynomially solvable subclasses except for Horn (however, it is renamable Horn). The results above for SLUR and q-Horn show that cycles in a random formula are abundant if $r > 4/(k(k-1))$. The next theorem shows that random formulas have few cycles if $r < .618/(k(k-1))$.

THEOREM 3.7. *Under $\mathcal{M}_{\Updownarrow,\backslash,||}$, $k \ge 3$, the average number of cycles in a random formula $\mathcal{F}$ is less than 1 when $r \equiv \frac{m}{n} < \frac{.618}{k(k-1)}$.*

Proof: We find the expected number of cycle patterns in $\mathcal{F}$ which is an over-estimate of the expected number of cycles in $\mathcal{F}$. The number of cycle patterns involving a sequence of x clauses is $m^{\overline{x}}n^{\overline{x}}$. The probability that a sequence of x clauses matches a pattern of x clauses is $\left(\binom{n-2}{k-2}/\binom{n}{k}\right)$. Therefore, the expected number of cycle patterns in $\mathcal{F}$ is

$$\sum_{x\geq 2} m^{\overline{x}}n^{\overline{x}}\left(\frac{\binom{n-2}{k-2}}{\binom{n}{k}}\right)^x < \sum_{x\geq 2}\left(\frac{mk(k-1)}{n-1}\right)^x < \left(\frac{mk(k-1)}{n-1}\right)^2\frac{1}{1-\frac{mk(k-1)}{n-1}}.$$

This is less than 1 if $mk(k-1)/n-1 < .618$. □

This result shows how closely SLUR, q-Horn, and other subclasses are tied to cycles in formulas: it seems that they are defeated rapidly by the presence of cycles (that is, as r rises, formulas are not SLUR, q-Horn, etc. soon after they begin to contain a significant number of cycles).

3.4. Easy Unsatisfiable Families of Formulas. Since it is based on unit resolution, one of the drawbacks of SLUR is it fails to provide a proof of unsatisfiability for all but some trivial unsatisfiable formulas. On the other hand, it is not hard to find non-trivial unsatisfiable formulas that are q-Horn. The question, whether q-Horn contains relatively many unsatisfiable formulas, seems to have the answer no from the results above since q-Horn formulas do not appear in abundance if $r > 4/(k(k-1))$ but formulas are satisfiable with high probability if $r < .25\ 2^k/k$. Indeed, the results above show that both SLUR and q-Horn are equally handicapped in solving unsatisfiable formulas. On top of this, it can be shown that there are families of unsatisfiable formulas that are easy to solve but are not in either SLUR or q-Horn. For example, the following result is proved in a forthcoming paper by Franco and Van Gelder.

THEOREM 3.8. *If* $\lim_{m,n\to\infty} m/n^{k-1+1/2^k} = \infty$, *a random formula* $\mathcal{F}$ *is unsatisfiable and can be solved in polynomial time with probability* $1-o(1)$.

4. Discussion and Conclusions

The aim of this paper is to determine the relative sizes of some well known polynomially solvable subclasses of Satisfiability. We used $\mathcal{M}_{\updownarrow,\backslash,||}$ and the ratio $r = m/n$ to provide a scale of formula "hardness" and determined where, on that scale, random formulas are members of the subclasses with high probability. We found that random formulas are not SLUR or q-Horn about where formula cycles begin to appear. Thus, neither subclass dominates in any range of r except where formulas are extremely "easy." The weakness of all the subclasses is that some local property can defeat them. In the case of SLUR and q-Horn this local property is the presence of cycles and we showed that both SLUR and q-Horn are about equally handicapped by this: that is, they are defeated by cycles that are similar in nature. This is surprising since, except for trivial cases, SLUR is useless on unsatisfiable formulas and q-Horn can solve non-trivial unsatisfiable formulas. Because of this, we had expected that q-Horn would dominate in some range of r where formulas are unsatisifable with high probability. But, this turned out not to be the case even though there is a range of r where unsatisfiable formulas are "easy."

We have also observed something unexpected about the Satisfiability index of (1)-(2) and [4]. According to this index, the subclass of formulas that satisfy a set of constraints with parameter $Z = 1 + c\frac{\log n}{n}$ is polynomially solvable and the subclass of formulas that satisfy these constraints with $Z = 1 + \frac{1}{n^\beta}$, for any $1 > \beta > 0$ is NP-complete. However, from Lemma 3.5 and Theorem 3.6, nearly all formulas satisfy the constraints with $Z > 1 + \frac{1}{n^\beta}$, for any $\beta > 0$, if $r > 4/(k(k-1)$ and in this range, up to $r = .25\ 2^k/k$, most random formulas are very easy but not usually in one of the well known polynomially solved subclasses of Satisfiability.

5. Acknowledgments

I wish to thank Allen Van Gelder for helpful comments on earlier drafts of this paper.

References

[1] B. Aspvall, M. F. Plass, and R. E. Tarjan, *A linear-time algorithm for testing the truth of certain quantified Boolean formulas.* Information Processing Letters **8**(3) (1979), 121–132.

[2] B. Aspvall, *Recognizing disguised NR(1) instances of the satisfiability problem.* Journal of Algorithms **1** (1983), 97–103.

[3] E. Boros, P. L. Hammer, and X. Sun, *Recognition of q-Horn formulae in linear time.* Discrete Applied Mathematics **55** (1994), 1–13.

[4] E. Boros, Y. Crama, P. L. Hammer, and M. Saks, *A complexity index for satisfiability problems.* SIAM Journal on Computing **23** (1994), 45–49.

[5] A. Z. Broder, A. M. Frieze, and E. Upfal, *On the satisfiability and maximum satisfiability of random 3-CNF formulas.* Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (1993), 322–330.

[6] V. Chandru and J. N. Hooker, *Extended Horn sets in propositional logic.* J. ACM **38** (1991), 205–221.

[7] M. T. Chao and J. Franco, *Probabilistic analysis of two heuristics for the 3-satisfiability problem.* SIAM J. on Computing **15** (1986), 1106–1118.

[8] M. T. Chao and J. Franco, *Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k satisfiable problem.* Information Sciences **51** (1990), 289–314.

[9] V. Chvátal and E. Szemerédi, *Many hard examples for resolution.* J. of ACM **35** (1988), 759–770.

[10] V. Chvátal and B. Reed, *Mick gets some (the odds are on his side).* Proc. 32nd Symposium on the Foundations of Computer Science (1992).

[11] M. Conforti, G. Cornuéjols, A. Kapoor, K. Vušković, and M. R. Rao, *Balanced Matrices.* Mathematical Programming: State of the Art. J. R. Birge and K. G. Murty, eds. Braun-Brumfield, United States. Produced in association with the 15th Int'l Symposium on Mathematical Programming, University of Michigan, 1994.

[12] M. Dalal, and D. W. Etherington, *A hierarchy of tractable satisfiability problems.* Information Processing letters **44** (1992), 173–180.

[13] W. F. Dowling and J. H. Gallier, *Linear-time algorithms for testing the satisfiability of propositional Horn formulae.* Journal of Logic Programming **1** (1984), 267–284.

[14] S. Even, A. Itai, and A. Shamir, *On the complexity of timetable and multi-commodity flow problems.* SIAM J. on Computing **5**(4) (1976), 691–703.

[15] J. Franco, J. Goldsmith, J. S. Schlipf, E. Speckenmeyer, R. P. Swaminathan, *An algorithm for the class of pure implicational formulas.* Proc. of the Workshop on Satisfiability, Siena, Italy (1996).

[16] A. M. Frieze, and S. Suen, *Analysis of two simple heuristics for random instances of 3-SAT.* Journal of Algorithms **20**(2) (1996), 312–355.

[17] G. Gallo, and M. G. Scutella, *Polynomially solvable satisfiability problems.* Information Processing Letters **29** (1988), 221–227.

[18] G. Gallo and D. Pretolani, *Hierarchies of polynomially solvable SAT problems.* Presented in the 3rd International Symposium on AI & Mathematics, (January, 1994).

[19] A. Goerdt, *A threshold for unsatisfiability.* Proc. 17th annual Symposium on Mathematical Foundations of Computer Science, Prague, Czechoslovakia, (1992).

[20] A. Itai and J. Makowsky, *On the complexity of Herbrand's theorem.* Working paper 243, Department of Computer Science, Israel Institute of Technology (1982).

[21] H. Kleine Büning, *On generalized Horn formulas and k resolution.* Theoretical Computer Science **116** (1993), 405–413.

[22] H. R. Lewis, *Renaming a set of clauses as a Horn set.* J. ACM **25** (1978), 134–135.

[23] J. S. Schlipf, F. Annexstein, J. Franco, and R. P. Swaminathan, *On finding solutions for extended Horn formulas.* Information Processing Letters **54** (1975), 133–137.

[24] M. G. Scutella, *A note on Dowling and Gallier's top-down algorithm for propositional Horn satisfiability.* Journal of Logic Programming **8** (1990), 265–273.

[25] K. Truemper, *Effective Logic Computation.* In preparation, expected 1997.

Computer Science, ECECS, University of Cincinnati, Cincinnati, OH 45221-0030
E-mail address: franco@gauss.ecess.uc.edu